IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of

Michael C. Stolowitz

Application No. 09/411,698

Filed: October 1, 1999

For: **ON-THE-FLY REDUNDANCY OPERATION IN THE SYNCHRONOUS DISK ARRAY CONTROLLER**

Date: December 22, 2000

Examiner: Thuan N. Du

Group Art Unit: 2782

AFFIDAVIT – RULE 1.132

I, Michael C. Stolowitz, being duly sworn, depose and say as follows:

1. I have a B.S. degree in Electrical Engineering from the University of California at Berkeley, granted in 1967.

2. I am a licensed Professional Engineer in the state of California (#CS2579) in control systems. I have more than 30 years of analog and digital circuit design experience, including 25 years as an independent consulting engineer. My client list includes Intel, Harris Digital Telephone, Memorex, National Semiconductor, Bell Northern Research, Bendix Flight Systems, Singer Link, Qume, the California Institute of Technology, Westland Aircraft in England, Siemens in Munich, and others.

3. I am the inventor or co-inventor of three U.S. patents.

4. I am the sole inventor of the patent application identified above.

5. Claim 1 of my application (as amended) is directed to a method of reading data from a RAID array of disk drives. One important feature of my invention is reconstruction of erroneous data "on the fly" during a read operation. By "on the fly," I mean is that read data is corrected as required during the transfer from the drives to the buffer.

6. Claim 1 reads as follows (with letters added for identifying individual elements or limitations):

Claim 1: (Amended) A method of reading striped digital data from a RAID array of disk drives, each drive having a respective data port of predetermined width coupled to an internal buffer, the method comprising the steps of:

(a) providing a single buffer memory having a data port coupled to all of the disk drive data ports for transferring digital data;

(b) providing a series of registers forming a common pipeline disposed in between the disk drive data ports and the buffer memory data port;

(c) providing a single address counter for addressing consecutive locations in the buffer memory;

(d) sending read commands to all of the disk drives so as to initiate read operations in all of the disk drives;

(e) waiting until read data elements are ready at all of the disk drive data ports;

(f) after read data elements are ready at all of the disk drive data ports, synchronously retrieving and storing the read data elements from all of the disk drive data ports into consecutive locations in the buffer memory under addressing control of the single address counter;

(g) wherein said synchronously retrieving and storing the read data elements from all of the disk drive data ports includes clocking the read data through the common pipeline so as to form a contiguous word serial data stream through the pipeline;

(h) concurrently computing redundant data from the read data while the read data moves through the pipeline;

(i) and, if a failed drive has been identified, substituting the computed redundant data into the word serial data stream in lieu of the failed disk drive data so as to form corrected read data; and

(j) storing the corrected read data into the buffer memory thereby providing the requested read data without incurring delay to reconstruct data stored on the failed disk drive and without storing erroneous data in the buffer memory.

7. Elements (d) (e) and (f) of claim 1 – the use of a common read strobe – effectively synchronizes the reading of data from the individual disk drives. When all the drives are ready, data is stored in the buffer as described in element (f). However, my method further calls for inserting the redundancy logic, i.e. the data reconstruction logic (formed using a pipeline of registers in the preferred embodiment), into this data path between the disk drive data ports and the buffer memory. This technique requires a single data stream. See element (g), *viz*: "said synchronously retrieving and storing the read data elements from all of the disk drive data ports includes clocking the read data through the common pipeline so as to form a contiguous word serial data stream through the pipeline..." [emphasis added].

8. Once I have formed a single, contiguous word-serial data stream through the pipeline, the method calls for (h) "concurrently computing redundant data from the read data while the read data moves through the pipeline." If a failed drive has been

identified, the computed redundant data is substituted into the data stream in lieu of the failed disk drive data. In this way, the correction is made on the fly. Moreover, the erroneous data (read from the failed disk drive) is never stored in the memory buffer.

9. Since only correct data is stored in the memory buffer, according to my invention, the buffer need only be sized to accommodate the total width the actual number of data drives, say N, rather than the total number of drives including the redundant drive, N+1. In a small system, where there are two "data drives" and one redundant drive, for example, this translates to a one-third smaller buffer memory size (2 versus 3) which is significant because the high-speed memory required for this type of buffer is relatively expensive.

10. I have personally reviewed U.S. Patent No. 5,765,186 ("Searby") to understand it's disclosure and teachings as one skilled in the art of electronic circuits of this type.

11. Searby's system transfers the disk read data from multiple drives to multiple buffers (whereas my claim 1 calls for using a single buffer memory.) The outputs of those buffers are then merged, according to Searby, to form a high-bandwidth data highway back to the host system.

12. The circuitry taught by Searby thus differs from my invention in that the delay shift register 57 of figure 3 is disposed between the RAM buffer memory 40 and the external data bus (register 50) which is connected to the host. This arrangement contrasts with the architecture of my invention, as illustrated in my figure 9, which shows that a shift register 902 is disposed between the disk drive data ports (900) and the buffer memory (in figure 9, "TO CACHE").

13. Searby thus inserts his redundancy logic as this data is merged, i.e. after it has been buffered. Accordingly, the Searby system requires buffering all the disk drive read data, *including the redundant data*. This is a small penalty – some 5% – in Searby's system because it contemplates 20 disk drives. My invention is directed to small RAID systems with, e.g., two drives plus one redundant drive. To buffer the redundant drive in that case, as taught by Searby, would represent a 50% increase in the buffer size, and a corresponding increase in memory cost.

14. The Examiner states that "it is a matter of design choice to dispose the pipeline in between the disk drive data ports and the buffer memory data ports." I respectfully disagree with that statement. The Examiner's proposed change to the Searby system would render it inoperative -- it wouldn't work. It is clear to me as one skilled in this art that the pipeline/data path elements are not commutative, for the following reasons:

15. In RAID systems, if N disc stores are required to meet the data capacity and bandwidth requirements, then N+1 disc stores are required to implement the checking and correcting system.

16. In the Searby patent, there is a buffering block consisting of N+1 data transfer means and a corresponding N+1 buffers. The buffering block inputs and outputs are each N+1 disc store data elements in width. For example, if there are five drives, each having a 16-bit data port, the buffer memory inputs and outputs would be 80 bits wide.

17. The buffer memory block, according to Searby, is followed by a redundancy block that takes in N+1 disc store data elements from the buffer (the 80-bit wide word in the above example), and outputs an N-disk store data element wide word (N=4 x 16 = 64-bit word) over a pipeline to the host system. Thus Searby's redundancy block (error correction logic) inputs one word size and outputs another. This is why the processing elements in this pipeline are not commutative; i.e., one cannot simply exchange the relative positions of the buffering and parity blocks in the pipeline because the word sizes are not compatible.

18. My invention has the following advantages over prior art:

(a) Memory Bandwidth: While RAM bandwidths are much greater than disk drive bandwidths, the total requirements for an array of drives and concurrent host transfers make the buffer bandwidth a critical resource. For a 3-drive application consisting of one redundant drive and two data drives, processing redundant data between the buffer and the host system (as taught by Searby) adds 50% to the buffer bandwidth requirements as 50% more data must be transferred in and out of the buffer memory for a given data throughput. In contradistinction, according to the present invention, no redundant data is stored in the buffer memory, since the redundancy and data correction logic is deployed between the buffer memory and the drives.

(b) Memory Capacity: If the buffer memory is going to provide some level of data caching, 50% more memory would be required to cache data plus redundancy for a 3-drive RAID system.

(c) Memory Scalability: The design and management of the buffer memory grows increasing complex as scalability is considered. For a 3-drive system, 1/3 of the memory is used to store redundant data and must be accessed concurrently with the data during host transfers. For a 5-drive system, 1/5 of the memory stores redundant data and must be accessed concurrently with host transfers. These inefficiencies are avoided by the methodology described in my claim 1.

19. My invention, as reflected in claim 1, is an overall read methodology. The on-the-fly data correction feature of my invention depends upon the other elements (a)

through (f) of the claim. These steps take care of synchronizing the drives to form the single, contiguous, word-serial data stream that, in turn, facilitates on-the-fly correction. Searby uses one buffer for each drive to collect the data and take care of synchronization. Once all the data is collected in the buffers, he can turn to data correction, but not on-the-fly as I have described.

20. I described the Searby type of system in my application at page 12, line 21, *viz*: "In prior art, because the disk transfers are asynchronous relative to one another, computation of the redundant check word had to wait until all the data was in the buffer memory. Each word of the buffer has to be read back for the purpose of the calculation. This doubles the number of accesses and required RAM buffer bandwidth for a given data rate. If less than double bandwidth is available, the data transfer rate will suffer."

21. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: 12/20/00

Michael C. Stolowitz
Danville, California

STATE OF CALIFORNIA)
) ss.
County of Contra Costa)

This 20th day of December, 2000, before me personally came the above-named **Michael C. Stolowitz**, who executed the foregoing instrument in my presence, and who acknowledged to me that he executed the same of his own free will for the purposes set forth herein.

Z. CINDY TSUNG
Commission # 1151498
Notary Public - California
Contra Costa County
My Comm. Expires Aug 15, 2001

Notary Public for California
My commission expires: August 15, 2001